SISTEMI OPERATIVI

Esercizio 1 del 16 Febbraio 1996

In un supermercato è presente un **banco di prodotti gastronomici**: tale banco viene gestito sempre da un solo **addetto** che serve in modo sequenziale i **clienti**.
Per effettuare un servizio ordinato, il banco prevede una gestione con numeri di prenotazione forniti da un apposito distributore automatico. I numeri vengono incrementati automaticamente, modulo **N**, con N sufficientemente grande in modo da non generare lo stesso numero più di una volta nella stessa giornata.
Ogni cliente, dopo aver preso il numero di prenotazione, si mette in attesa di essere servito: quando è il suo turno, viene servito dall'addetto effettuando l'acquisto e quindi se ne va. L'addetto, quindi, serve un cliente per volta, altrimenti rimane in attesa.
Si gestisca la politica di gestione del banco di gastronomia facendo uso del costrutto monitor: si giustifichino le scelte fatte commentando adeguatamente il codice e di descriva la struttura dei processi cliente e addetto.

Facoltativamente: Si discuta se nella politica implementata sono possibili casi di starvation e/o deadlock e, in caso affermativo si propongano soluzioni per eliminarli.

```
program supermercato

const N = ...; { numero di biglietti }

type cliente = process
begin
      banco.ordina;
      <prendi il pacco e esci>
end

type addetto = process
begin
      repeat { in questo caso è necessario un ciclo infinito }
            banco.servi;
            <serve il cliente>
            banco.termina_servizio;
      until false
end

type banco_gastronomico = monitor

{ variabili del monitor }
var  servito: 1..N;
      { numero del cliente servito }
      biglietto: 1..N;
      { numero del foglietto disponibile }
      coda_clienti: condition;
      { coda su cui sospendere i clienti che aspettano di
      essere serviti }
      coda_addetto: condition;
      { coda su cui sospendere l'addetto in attesa di clienti }
      coda_servito: condition;
      { coda su cui sospendere il cliente in attesa del pacco }
```

```
procedure entry ordina
begin
        mio_numero := biglietto;
        biglietto := (biglietto + 1) mod N;
        while servito <> mio_numero do
        { mi sospendo se non è il mio turno }
                coda_clienti.wait;

        { se l'addetto è sospeso, viene risvegliato }
        { se invece è già sveglio, la signal non ha effetto }
        coda_addetto.signal;
        { mi sospendo in attesa del pacco }
        coda_servito.wait;
end

procedure entry servi
begin
        if not coda_clienti.queue
        then
                coda_addetto.wait;
        else
                coda_clienti.signal;
end

procedure entry termina_servizio
var i : integer;
begin
        coda_servito.signal;
        servito := (servito + 1) mod N ;
        if coda_clienti.queue
        then
                for i := servito to biglietto do
                        coda_clienti.signal;
end
```

```
begin { inizializzazione delle variabili }
      servito := 1;
      biglietto := 1;
end

var  banco: banco_gastronomico; { il nostro monitor }
      cl1, cl2, ... : cliente;
      a : addetto;

begin end.
```

Considerazioni

Il risveglio dei camerieri/clienti deve essere fatto in alternativa alla sospensione, poiché, non sapendo chi dei due risveglia l'altro, fare due signal comprometterebbe la correttezza della soluzione.

Starvation

La soluzione proposta non presenta starvation, infatti ogni cliente viene servito secondo l'ordine con cui arriva.